EMIDE Application Development Kit

--- USER MANUAL

Liu Xing, SMIR team, LIMOS UMR 6158 CNRS
University Blaise Pascal Clermont-Ferrand
Version 1, June 2013

Summary:

I. Introduction to the iLive Platform 3

II. Background of the mid-layer software EMIDE 3

III. About the EMIDE-Kit 4

IV. The WSN network established by the iLive 4

V. Demo Projects in the EMIDE-Kit 5

VI. Application Image Size 5

VII. Contact and feedback 6

I. Introduction to the iLive Platform

iLive is a wireless sensor network platform dedicated to the environmental data collection applications. It is developed by the SMIR team of LIMOS UMR 6158 CNRS laboratory.

iLive node is depicted in the Figure 1. Its key features are as follows:

- Low energy consumption, e.g., 45 mA in active mode and 5-6 µA in sleep mode;
- Powered by two AA standard batteries (1.5 V). An iLive node equipped with 4 Watermark sensors and sampling once per day can have a lifetime of 5 years;
- Dimension : 76mm×40mm ;
- 4 Watermark sensors ;
- 3 Decagon sensors ;
- 1 temperature sensor ;
- 1 air humidity sensor ;
- 1 light sensor ;
- 1 RS232/USB slave port ;
- ZigBee (IEEE802.15.4) wireless access medium.

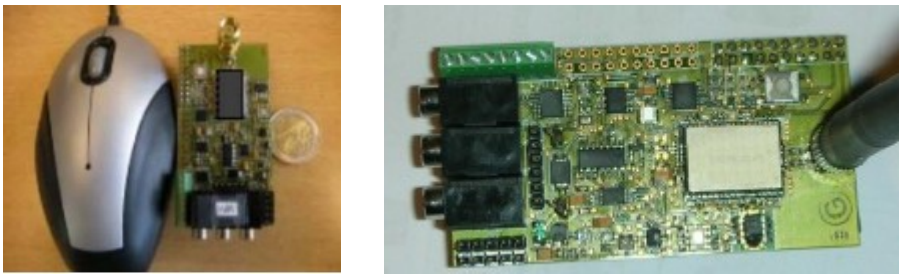

Figure 1. iLive WSN Node.

II. Background of the mid-layer software EMIDE

Programming and reprogramming are two important processes for application development in WSN. Currently, these processes are difficult for the WSN users. On one hand, the hardware platforms in the WSN are diverse, thus the users need to understand the diverse low-level hardware details for the application programming. On the other hand, most WSN nodes are prone to be deployed in the harsh environments where humans cannot access, thus the WSN reprogramming needs to be done remotely through the wireless. Yet, the wireless transmission in the WSN is high in the energy consumption and limited in the communication bandwidth. Consequently, the WSN reprogramming process becomes complicated.

One way to address the challenges above is to decouple the application from the low-level system and provide a set of services in the system space for the application programming. By this way, two separated images will be generated: the application image and the system image. The system image can be built by the WSN experts and pre-burned to the WSN nodes. And

then, the WSN users only need to focus in the application space and upload the application image to the target nodes when necessary. By this means, the application programming can be simplified as the users no more need to understand the low-level system details. Moreover, the reprogramming performance can be improved as only the application image other than the monolithic software image is needed to be updated.

To decouple the user application from the WSN system, the mid-layer software EMIDE is developed. Currently, EMIDE can support the development of the multitasking pre-linked machine-code applications, act as a bridge between the application space and the system space, and is efficient in both the memory consumption and the energy cost.

III. About the EMIDE-Kit

With the EMIDE, the WSN application is decoupled from the WSN system, and can be built independently. In order to provide a user-friendly EMIDE application development environment to the WSN users, the 1st version EMIDE application development kit (*EMIDE-Kit*) is distributed.

Thanks to the EMIDE-Kit, the users can program the WSN application easily by the C language, and reprogram the new application image to the WSN nodes simply by updating only a minor size code (commonly 100-300 bytes, the updating can be completed quickly within tens of seconds, and less energy will be consumed during the reprogramming process).

IV. The WSN network established by the iLive

The network established by the iLive nodes follows the IEEE802.15.4 ZigBee specification [1]. In this network, two kinds of nodes exist: the coordinator and the end-devices. The coordinator is responsible for the new network establishment, the devices associations as well as the communications with the external network. The end-devices are in charge of collecting the sensor data and transmitting the data packets to the Coordinator. All the WSN nodes can be controlled remotely from Internet by the user terminal devices, such as laptop, PDA, mobile phone, etc, depicted in Figure 2.

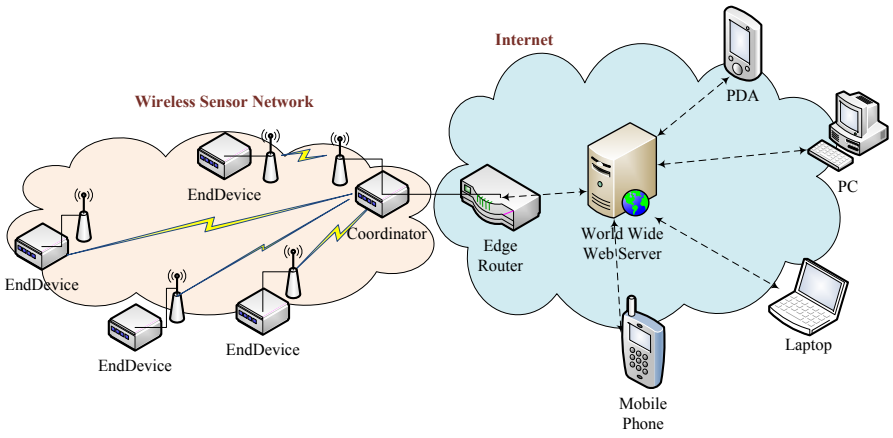

Figure 2. Sensor Network established by the iLive nodes.

V. Demo Projects in the EMIDE-Kit

After a WSN network is established, it is essential to evaluate the network *packet loss rate* (*PLR*) and the *throughput*. In the EMIDE-Kit, two application projects are provided to evaluate these performances. These projects locate under the directory "EMIDE-Kit/projects", and are named as the "pkt_lost_rate_test" and the "throughput_test" respectively.

The *PLR* project is implemented as follows: the coordinator node acts as the sender and the end-device nodes act as the receivers. Periodically, the sender broadcasts a wireless packet. After the receivers receive this packet, they will send back a packet simultaneously to the sender (Figure 3). According to the number of the receiver nodes (*Nsend*) and the number of the sender received packets (*Nrcv*), the *PLR* can be calculated:

$$PLR = Nrcv / Nsend.$$

Two directories exist in the *PLR* project, one is for the *coordinator* node and the other is for the *end-device* nodes. In each directory, an executable binary "*EMIDE.exe*" exists. This binary can be executed by just clicking it. After executed, the application program (named the "*app_coordinator.c*" or the "*app_enddevice.c*") will be processed (be built, be pre-linked, etc.), and then a final executable application image will be generated (named "*wsn_app.bin*"). This image will then be reprogrammed to the WSN nodes through the wireless.

To run the "*EMIDE.exe*", the AVR GNU compiler needs to be installed, and it can be downloaded from the website: http://winavr.sourceforge.net/.

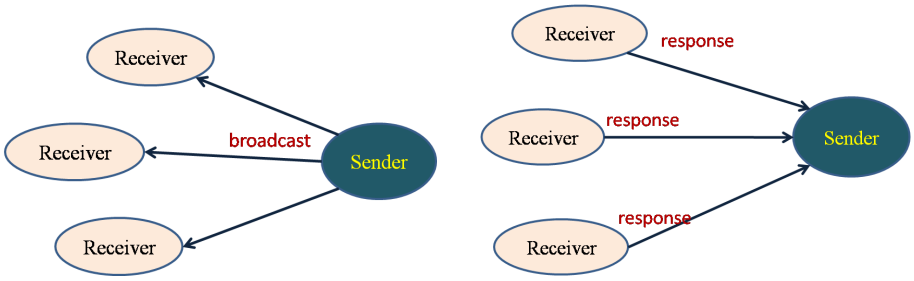

Figure 3. Measurement of the packet lost rate in the WSN.

VI. Application Image Size

The application sizes of the EMIDE applications in these projects are shown in the Table 1. From this table, it can be seen that the code size of the WSN application becomes smaller after EIMDE is applied. Thus, the WSN reprogramming performance can be improved greatly, e.g., only 2 to 3 ZigBee packets need to be transmitted during the reprogramming process, this process will be short in the time cost and low in the energy cost.

Projects	Directory	Node type	Source program	Final executable image size	Wireless transmission frames during the reprogramming process
PLR project	projects/ pkt_lost_rate_test	Coordinator	app_coordinator.c	302 bytes	Only 3 frames for updating
		End-device	app_enddevice.c	186 bytes	2 frames
Throughput project	projects/ throughput_test	Coordinator	app_coordinator.c	312 bytes	3 frames
		End-device	app_enddevice.c	186 bytes	2 frames

Table 1. Reprogramming sizes of the EMIDE applications.

VII. Contact and feedback

Any question or suggestion will be welcome to the email: liu@isima.fr.

Xing Liu

June the 16th, 2013

Clermont-Ferrand, France